**Practice 1:**

* There are four coding errors in the following statement. Can you identify them?

SELECT employee\_id, last\_name

sal x 12 ANNUAL SALARY

FROM employees;

Solution:

1. Missing , after last\_name
2. Sal >> salary
3. X >> \*
4. ANNUAL SALARY >> " ANNUAL SALARY"

The correct statement is:

(SELECT employee\_id, last\_name, salary\*12 "ANNUAL SALARY" FROM employees;)

* Determine the structure of the EMPLOYEES table.

Solution:

Describe employees ;

* The HR department has requested a report of all employees and their job IDs. Display the last name concatenated with the job ID (separated by a comma and space) and name the column Employee and Title.

Solution:

select last\_name ||q'# , #'||job\_id as "Employee and Title" from employees ;

* The HR department wants a query to display all unique job IDs from the EMPLOYEES table.

Solution:

select distinct job\_id from employees;

* To familiarize yourself with the data in the EMPLOYEES table, create a query to display all the data from that table. Separate each column output by a comma. Name the column title THE\_OUTPUT.

Solution:

Select employee\_id || ',' || first\_name || ',' || last\_name || ',' || email || ',' || phone\_number || ',' || hire\_date || ',' || job\_id || ',' || salary || ',' || commission\_pct the\_output from employees as THE\_OUTPUT;

Practice 2

The HR department needs your assistance with creating some queries.

1. Due to budget issues, the HR department needs a report that displays the last name and salary of employees who earn more than $12,000.

**Solution**

select last\_name, salary

from employees

where salary > 12000;

1. Create a report that displays the last name and department number for employee number 176.

**Solution**

select last\_name, department\_id

from employees

where employee\_id = 176;

1. The HR departments needs to find high-salary and low-salary employees. Display the last name and salary for any employee whose salary is not in the range of $5,000 to $12,000.

**Solution**

select last\_name, salary

from employees

where salary>12000 or salary<5000;

1. Create a report to display the last name, job ID, and start date for the employees with the last names of Matos and Taylor. Order the query in ascending order by start date.

**Solution**

select last\_name, job\_id, hire\_date

from employees

where last\_name='Matos' or last\_name='Taylor'

order by hire\_date;

1. Display the last name and department number of all employees in departments 20 or 50 in ascending alphabetical order by name.

**Solution**

select last\_name, department\_id

from employees

where department\_id in (20,50)

order by last\_name;

1. Display the last name and salary of employees who earn between $5,000 and $12,000 and are in department 20 or 50. Label the columns Employee and Monthly Salary, respectively.

**Solution**

select last\_name "Employee", salary "Monthly Salary"

from employees

where (salary between 5000 and 12000) and department\_id in (20,50);

1. The HR department needs a report that displays the last name and hire date for all employees who were hired in 1994.

**Solution**

select last\_name, hire\_date

from employees

where hire\_date like '%94';

1. Create a report to display the last name and job title of all employees who do not have a manager.

**Solution**

select last\_name, job\_id

from employees

where manager\_id is null;

1. Create a report to display the last name, salary, and commission of all employees who earn commissions. Sort data in descending order of salary and commissions.

**Solution**

select last\_name, salary, commission\_pct

from employees

where commission\_pct is not null

order by salary, commission\_pct;

1. Display all employee last names in which the third letter of the name is *a.*

**Solution**

select last\_name

  from employees

  where last\_name like '\_\_a%';

1. Display the last name of all employees who have both an *a* and an *e* in their last name.

**Solution**

select last\_name

from employees

where last\_name like '%a%' and last\_name like '%e%';

1. Display the last name, job, and salary for all employees whose job is sales representative (SA\_REP)or stock clerk (ST\_CLERK)and whose salary is not equal to $2,500, $3,500, or $7,000.

**Solution**

select last\_name, job\_id, salary

from employees

where (job\_id='SA\_REP' or job\_id='ST\_CLERK') and salary not in(2500,3500,7000);

1. Display the last name, salary, and commission for all employees whose commission amount is 20%.

**Solution**

select last\_name "Employee", salary "Monthly Salary", commission\_pct

from employees

where commission\_pct=0.2;

Practice 3

1. Write a query to display the current date. Label the column Date.  
   SOLUTION  
   SELECT SYSDATE AS "Date" FROM DUAL;
2. The HR department needs a report to display the employee number, last name, salary, and salary increased by 15.5% (expressed as a whole number) for each employee. Label the column New Salary. Place your SQL statement in a text file named lab\_03\_02.sql.  
   SOLUTION  
   select EMPLOYEE\_ID, LAST\_NAME, SALARY, SALARY+(SALARY\*15.5/100) AS "New Salary"  
   from employees;
3. Modify your query lab\_03\_02.sql to add a column that subtracts the old salary from the new salary. Label the column Increase. Save the contents of the file as lab\_03\_04.sql. Run the revised query.   
   SOLUTION  
   select EMPLOYEE\_ID, LAST\_NAME, SALARY, salary+(salary\*15.5/100) "New Salary",(salary+(salary\*15.5/100))-salary "Increase" from employees;
4. Write a query that displays the last name (with the first letter uppercase and all other letters lowercase) and the length of the last name for all employees whose name starts with the letters J, A, or M. Give each column an appropriate label. Sort the results by the employees’ last names.  
   SOLUTION  
   select initcap(last\_name) "Name", length(last\_name) "Length of Name"  
   from employees  
   where last\_name like 'J%' or last\_name like 'A%' or last\_name like 'M%'   
   order by last\_name;
5. The HR department wants to find the length of employment for each employee. For each employee, display the last name and calculate the number of months between today and the date on which the employee was hired. Label the column MONTHS\_WORKED. Order your results by the number of months employed. Round the number of months up to the closest whole number. Note: Your results will differ.  
   SOLUTION  
   select last\_name, round(months\_between(sysdate,hire\_date),0) Months\_worked   
   from employees   
   order by 2;
6. Create a query to display the last name and salary for all employees. Format the salary to be 15 characters long, left-padded with the $ symbol. Label the column SALARY.  
   SOLUTION  
   select last\_name, lpad(salary,15,'$') Salary  
   from employees;

Practice 4

1. Display the last name, hire date, and day of the week on which the employee started. Label the column DAY. Order the results by the day of the week**.**SOLUTION  
   select last\_name,hire\_date,TO\_CHAR(hire\_date,'DAY') AS DAY  
   from employees  
   order by hire\_date, DAY;
2. Create a query that displays the employees’ last names and commission amounts. If an employee does not earn commission, show “No Commission.” Label the column COMM.  
   SOLUTION   
   select last\_name,   
   NVL2(commission\_pct, TO\_CHAR(commission\_pct), 'no commission') "COMM"  
   from employees;

1. Using the DECODE function, write a query that displays the grade of all employees based on the value of the column JOB\_ID, using the following data:

Job Grade

AD\_PRES A

ST\_MAN B

IT\_PROG C

SA\_REP D

ST\_CLERK E

None of the above 0

SOLUTION  
select job\_id, DECODE(JOB\_ID,  
 'AD\_PRES','A',   
'ST\_MAN', 'B',  
'IT\_PROG','C',   
'SA\_REP','D',   
'ST\_CLERK','E', 0) "GRADE"  
from employees;

1. Rewrite the statement in the preceding exercise using the CASE syntax.  
   SOLUTION  
   SELECT job\_id, CASE job\_id   
   WHEN 'ST\_CLERK' THEN 'E'   
   WHEN 'SA\_REP' THEN 'D'   
   WHEN 'IT\_PROG' THEN 'C'   
   WHEN 'ST\_MAN' THEN 'B'   
   WHEN 'AD\_PRES' THEN 'A'   
   ELSE '0' END GRADE  
   FROM employees;

Practice 5

Determine the validity of the following three statements. Circle either True or False.

1. Group functions work across many rows to produce one result per group.  
 True/False  
SOLUTION: **TRUE**

2. Group functions include nulls in calculations.  
 True/False  
SOLUTION: **FALSE**

1. The WHERE clause restricts rows prior to inclusion in a group calculation.  
    True/False  
   SOLUTION: **TRUE**

The HR department needs the following reports:

1. Find the highest, lowest, sum, and average salary of all employees. Label the columns

Maximum, Minimum, Sum, and Average, respectively. Round your results to the nearest whole number.

SOLUTION  
SELECT ROUND(MAX(salary),0) "Maximum",

ROUND(MIN(salary),0) "Minimum",

ROUND(SUM(salary),0) "Sum",

ROUND(AVG(salary),0) "Average"

FROM employees;

5. Display the minimum, maximum, sum, and average salary for each job type.

SOLUTION  
SELECT job\_id, ROUND(MAX(salary),0) "Maximum",   
ROUND(MIN(salary),0) "Minimum",   
ROUND(SUM(salary),0) "Sum",   
ROUND(AVG(salary),0) "Average"   
FROM employees;   
GROUP BY job\_id;

1. Write a query to display the number of people with the same job.

SOLUTION  
SELECT job\_id, COUNT(\*)   
FROM employees   
GROUP BY job\_id;

1. Determine the number of managers without listing them. Label the column Number of Managers.

*Hint: Use the MANAGER\_ID column to determine the number of managers.*

SOLUTION  
SELECT COUNT(DISTINCT manager\_id) "Number of Managers"   
FROM employees;

1. Find the difference between the highest and lowest salaries. Label the column DIFFERENCE.

SOLUTION  
SELECT MAX(salary) - MIN(salary) AS "DIFFERENCE"   
FROM employees;

1. Create a report to display the manager number and the salary of the lowest-paid employee for that manager. Exclude anyone whose manager is not known. Exclude any groups where the minimum salary is $6,000 or less. Sort the output in descending order of salary.

SOLUTION  
SELECT manager\_id, MIN(salary)   
FROM employees   
WHERE manager\_id IS NOT NULL   
GROUP BY manager\_id   
HAVING MIN(salary) > 6000   
ORDER BY MIN(salary) DESC;

Practice 6

1. Write a query for the HR department to produce the addresses of all the departments. Use the LOCATIONS and COUNTRIES tables. Show the location ID, street address, city, state or province, and country in the output. Use a NATURAL JOIN to produce the results.

Solution

SELECT location\_id, street\_address, city, state\_province, country\_name

FROM locations

NATURAL JOIN countries;

1. The HR department needs a report of all employees. Write a query to display the last name, department number, and department name for all employees.  
   Solution

SELECT last\_name, department\_id, department\_name   
FROM employees   
JOIN departments   
USING (department\_id);

1. The HR department needs a report of employees in Toronto. Display the last name, job, department number, and department name for all employees who work in Toronto.

Solution

SELECT e.last\_name, e.job\_id, e.department\_id, d.department\_name   
FROM employees e JOIN departments d

ON(e.department\_id = d.department\_id)

JOIN locations l

ON(d.location\_id = l.location\_id)

WHERE LOWER(l.city) = 'Toronto';

1. Create a report to display employees’ last name and employee number along with their manager’s last name and manager number. Label the columns Employee, Emp#, Manager, and Mgr#, respectively.

Solution

SELECT w.last\_name "Employee", w.employee\_id "EMP#", m.last\_name "Manager", m.employee\_id "Mgr#"

FROM employees w join employees m

ON (w.manager\_id = m.employee\_id);

1. Modify SQL\_Statement#4 to display all employees including King, who has no manager. Order the results by the employee number.

Solution

SELECT w.last\_name "Employee", w.employee\_id "EMP#", m.last\_name "Manager", m.employee\_id "Mgr#"

FROM employees w

LEFT OUTER JOIN employees m

ON (w.manager\_id = m.employee\_id) ORDER BY 2;

Practice 7

1. Create a report that displays the employee number, last name, and salary of all employees who earn more than the average salary. Sort the results in order of ascending salary.

**SOLUTION**

SELECT employee\_id, last\_name, salary   
FROM employees   
WHERE salary > (SELECT AVG(salary)   
 FROM employees)   
ORDER BY salary;

1. Write a query that displays the employee number and last name of all employees who work in a department with any employee whose last name contains a *u*.

**SOLUTION**

SELECT employee\_id, last\_name   
FROM employees   
WHERE department\_id IN (SELECT department\_id   
 FROM employees   
WHERE last\_name like '%u%');

1. The HR department needs a report that displays the last name, department number, and job ID of all employees whose department location ID is 1700.

**SOLUTION**

SELECT last\_name, department\_id, job\_id   
FROM employees   
WHERE department\_id IN (SELECT department\_id   
 FROM departments   
WHERE location\_id = 1700);

1. Create a report for HR that displays the last name and salary of every employee who reports to King.

**SOLUTION**

SELECT last\_name, salary   
FROM employees   
WHERE manager\_id = (SELECT employee\_id   
 FROM employees   
 WHERE last\_name = 'King');

1. Create a report for HR that displays the department number, last name, and job ID for every employee in the Executive department.

**SOLUTION**

SELECT department\_id, last\_name, job\_id   
FROM employees   
WHERE department\_id IN (SELECT department\_id   
 FROM departments   
 WHERE department\_name = 'Executive');

1. Display the employee number, last name, and salary of all employees who earn more than the average salary and who work in a department with any employee whose last name contains a “*u”*.

**SOLUTION**

SELECT employee\_id, last\_name, salary   
FROM employees   
WHERE department\_id IN (SELECT department\_id   
 FROM employees   
 WHERE last\_name like '%u%')   
AND salary > (SELECT AVG(salary)   
 FROM employees);